RECEIVED
CENTRAL FAX CENTER
JUN 2 8 2005

Certificate of Transmission under 37 CFR 1.8

I hereby certify that this correspondence is being facsimile transmitted to the United States Patent and Trademark Office

on June 28, 2005.
Date

Signature

William E. Curry
Typed or printed name of person signing Certificate

43,572	202.220.4200
Registration Number if applicable	Telephone Number

Note: Each paper must have its own certificate of transmission, or this certificate must identify each submitted paper.

Inventor:	Stephan J. JOURDAN et al..
Serial No.:	09/608,624
Filed:	June 30, 2000
Title:	TRACE INDEXING VIA TRACE END ADDRESSES
Examiner:	Henry TSAI
Group:	2183

Appeal Brief (3 copies, 33 pgs.) 2207/8609 June 28, 2005 WEC/ske

Application Ser. No. 09/608,624
Attorney Docket No. 2207/8609

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED
CENTRAL FAX CENTER
JUN 2 8 2005

INVENTORS: Stephan L. JOURDAN, et al.

SERIAL NO: 09/608,624

FILING DATE: June 30, 2000

TITLE: TRACE INDEXING VIA TRACE END ADDRESSES

ART UNIT: 2183

EXAMINER: Henry TSAI

Mail Stop Appeal Brief - Patents
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

SIR:

This is a brief in support of an appeal filed in the above-identified application.

I. Real Party In Interest

The real party in interest is the Assignee, Intel Corporation.

II. Related Appeals and Interferences

There are no other appeals or interferences known to Appellant, Appellant's legal representative, or Assignee which will directly affect or be directly affected by or havo a bearing on the Board's decision in this Appeal.

III. Status of Claims

The application as filed included claims 1-19. Claims 20-27 were added in an amendment filed September 4, 2003. Claims 8 and 21 were canceled and claims 28-37 were amended in an amendment filed April 8, 2004. Claims 31-37 were canceled and claims 38-43 were added in an amendment filed October 4, 2004. Claims 4-7, 9-15, 17-19, 23-27 and 41-43 are allowed. Claims 3, 22, 30 and 40 are objected to. Claims 1, 2, 16, 20, 28, 29, 38 and 39 have been finally rejected and are herein on appeal.

IV. Status of Amendments

The amendments filed March 4, 2005 in response to the final rejection mailed January 4, 2005 have been entered.

V. Summary of Claimed Subject Matter

(References are to the specification and drawings.)

Independent claim 1.

Apparatus, comprising a memory entry storing a trace having a multiple-entry, single exit architecture.

Explanation.

Embodiments of the present invention relate to an "extended block" architecture for instructions in a processor. The extended blocks are types of "traces." An extended block, according to embodiments of the present invention, may have multiple entry points but only a single exit point. (Page 2, lines 23-25.) Among other advantages, the traces provide for lessened redundancy, may be dynamically extended and may include shared blocks. (Page 3, lines 1-6.) The traces may be stored in a memory, such as a block cache 280. (Page 3, line 20 and FIG. 2.)

Examples of traces according to embodiments of the present invention are illustrated in FIGs. 4-6. As can be seen in these figures, the traces may have multiple

entry points but only a single exit point. For example, FIG. 4(c) shows a trace with two entry points IP_3 and IP_4, with a single exit point IP_1.

Independent claim 16.

A processing engine, comprising:

a front end stage storing blocks of instructions in a multiple-entry, single exit architecture when considered according to program flow, and

an execution unit in communication with the front end stage.

Explanation.

Instruction blocks having a multiple-entry, single exit architecture have been discussed above. As further discussed above, the instruction blocks may be stored in a memory, which in turn may be part of a processor front end stage. In particular, the memory may be a block cache 280 that is part of an extended block cache ("XBC") 220 that communicates with an execution unit. (Page 3, lines 7-8 , line 20, line 26 and FIG. 2.)

Independent claim 20.

Apparatus, comprising a memory entry storing a sequence of program instructions as a trace, the instructions defining a program flow that progresses from any instruction therein to a last instruction in the memory entry and in which the trace has multiple separate prefixes.

Explanation.

A trace may be a sequence of instructions in program order. (Page 1, lines 17-19.) According to embodiments of the present invention, the sequence may have different prefixes but a common suffix, so that all of the prefixes lead to a common last instruction. (Page 6, lines 5-7 and FIG. 4(c).

Independent claim 28.

A trace, comprising a sequence of program instructions assembled in order according to program flow, the sequence having a multiple-entry, single exit architecture.

Explanation.

A trace may be a sequence of instructions in program order. (Page 1, lines 17-19.) According to embodiments of the present invention, the sequence may have a multiple-entry, single exit architecture. (Page 2, lines 23-25.)

Independent claim 38.

Apparatus, comprising:

a memory having at least one memory entry; and

the at least one memory entry to store a trace having a multiple-entry, single exit architecture.

Explanation.

A trace may be a sequence of instructions in program order. (Page 1, lines 17-19.) According to embodiments of the present invention, the sequence may have a multiple-entry, single exit architecture. (Page 2, lines 23-25.) The traces may be stored in a memory, such as a block cache 280. (Page 3, line 20 and FIG. 2.)

VI. Grounds of Rejection to be Reviewed on Appeal

Claims 1, 2, 16, 20, 28, 38 and 39 were rejected under 35 USC 102(e) as being anticipated by Agarwal (US 5,966,541).

VII. Argument

Agarwal does not anticipate the rejected claims.

To anticipate a claim under § 102, a single prior art reference must identically disclose each and every claim element. See Lindeman Machinenfabrik v. American Hoist and Derrick, 730 F.2d 1452, 1458 (Fed. Cir. 1984). If any claimed element is absent from a prior art reference, it cannot anticipate the claim. See Rowe v. Dror, 112 F.3d 473, 478 (Fed. Cir. 1997). In view of the foregoing authority, the Applicant respectfully submits that the cited reference does not support the asserted rejection.

Independent claim 1.

Discussion begins with independent claim 1. Agarwal does not anticipate claim 1 for at least the reason that Agarwal does not disclose a memory ***entry*** that stores a trace having a multiple-entry, single exit architecture as recited in claim 1.

Agarwal contains no disclosure of a relationship between a memory entry and a trace. Instead, Agarwal's disclosure relates to software generally, and more particularly to an arrangement for "patching" faulty software, especially for the Year 2000 problem. In view of this, as an equivalent to the claimed memory entry, the Examiner can offer only "the space containing blocks 101, 102 and 103 as shown in Fig. 8" (final Office Action, item 5, lines 4-6). But this is hardly adequate; claim 1 requires that the trace be stored in a ***memory entry***, not merely memory generally. Agarwal does not disclose this subject matter. Agarwal describes no relationship between his blocks and memory, whether they fit in single entries or whether they are distributed across multiple memories. This isn't surprising since Agarwal is devoted to finding solutions to the Year 2000 problem – any relationship between his blocks 101-103 and memory entries is immaterial to the performance of his processes.

Further, it is observed that Agarwal does not mention traces at all. A trace is a specific code structure that occurs in processors. This structure cannot be found in Agarwal. Notwithstanding, the Examiner alleges that blocks 101-103 shown in FIG. 8 of Agarwal are equivalent to the claimed trace. See the final Office Action, lines 8-9: "... storing a trace (including blocks 101, 102 and 103 as shown in Fig. 8)". However, even assuming solely for purposes of argument that Agarwal's blocks 101, 102 and 103 correspond to the claimed trace, the blocks 101-103 at best have a *single-entry, single-exit* architecture. This is evident from FIG. 8 of Agarwal, an annotated version of which is shown below.

Application Ser. No. 09/608,624
Attorney Docket No. 2207/8609

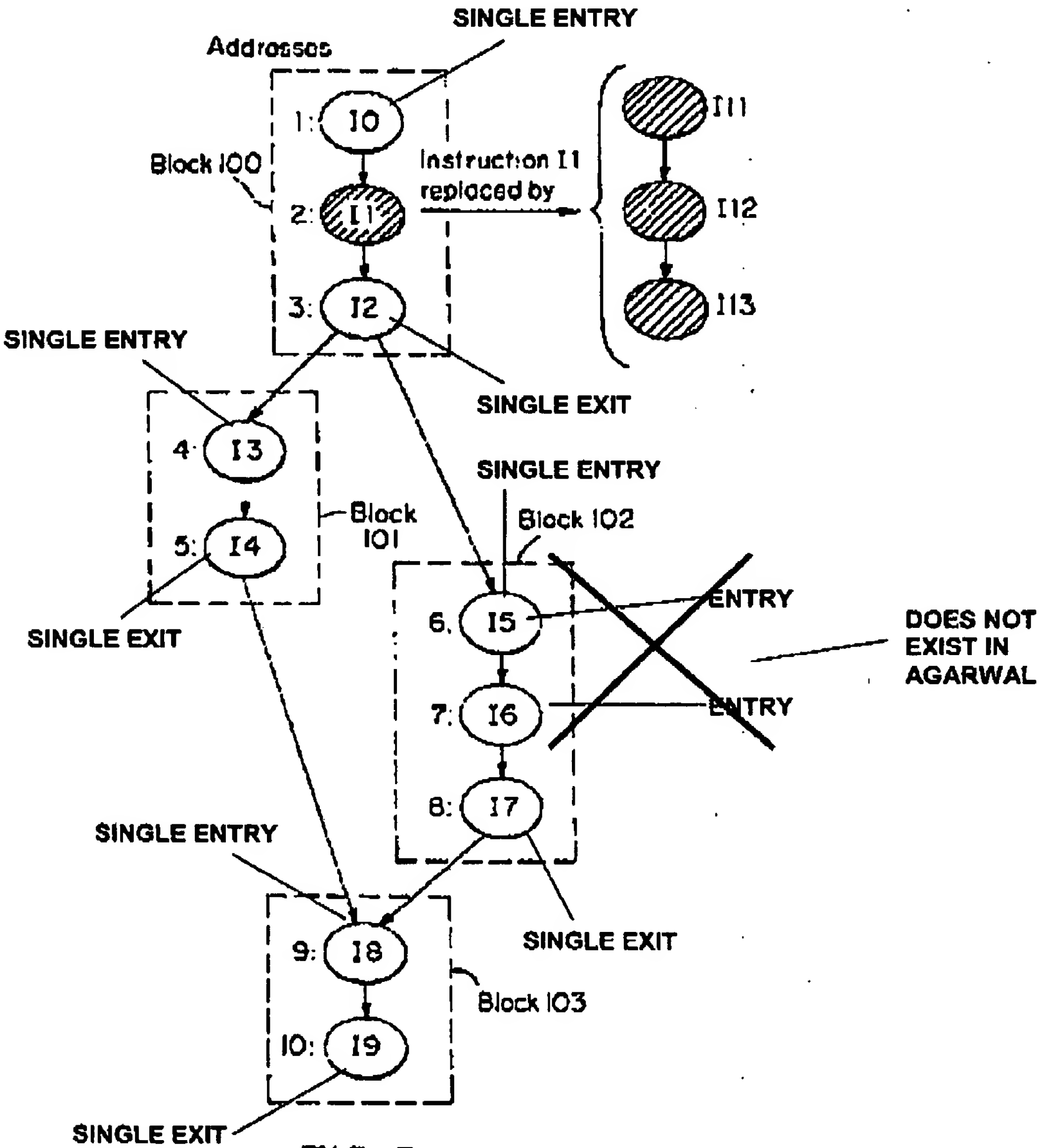


FIG. 8
(Agarwal, US 5,966,541)

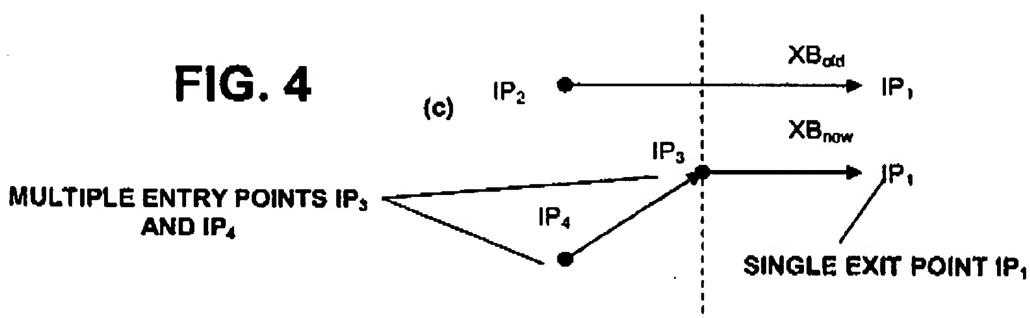
In the annotated FIG. 8, it can be seen that Agarwal dissects the code into separate blocks 100, 101, 102 and 103, each of which have a single entry, single exit architecture. Specifically, block 100 has a single entry point at I0 and a single exit point

at I2, block 101 has a single entry point at I3 and a single exit point at I4, and so on. Nowhere do the blocks exhibit a multiple entry structure, as illustrated hypothetically on block 102. Agarwal offers no teaching or suggestion to consider the code any other way.

In light of the above, it is apparent that the Examiner's strained effort to find multiple entry points in Agarwal's blocks 100-103 falls short. The Examiner refers specifically to "I2 to I3 and from I2 to I5" as multiple entry points (e.g., in the final Office Action, item 5, lines 9-10). However, under Agarwal's scheme, clearly I2 is not an entry, but an exit.

An annotated FIG. 4(c) of the present application, below, shows an example of the claimed multiple-entry, single-exit architecture by way of contrast to Agarwal's blocks 100-103. In FIG. 4(c), a trace has multiple entry points IP_3 and IP_4, and a single exit point IP_1.

FIG. 4



In view of the foregoing, claim 1 is allowable over Agarwal.

Dependent claims 2 and 3.

Rejected claim 2 depends on claim 1 and is therefore likewise allowable over Agarwal for at least the reasons discussed in connection with claim 1. Further, the objection to claim 3 should be withdrawn since claim 3 depends on allowable claim 1.

Independent claim 16.

Agarwal does not anticipate claim 16 for at least the reason that Agarwal does not disclose a front end stage storing blocks of instructions in a multiple-entry, single exit architecture when considered according to program flow. Moreover, Agarwal is completely silent as to a front end stage storing such blocks, and as to an execution unit in communication with the front end stage.

Agarwal's deficiencies concerning the multiple-entry, single exit architecture of embodiments of the present invention have been discussed above. As to the claimed front end stage in communication with an execution unit, Agarwal discloses only software structures with any specificity. Disclosure of hardware goes no deeper than "some memory" (col. 11, lines 36-38). Accordingly, Agarwal cannot meet the recitations of claim 16. Claim 16 is therefore allowable over Agarwal.

Independent claim 20.

Agarwal does not anticipate claim 20 for at least the reason that Agarwal does not disclose a trace with multiple separate prefixes as recited. This feature is analogous to the multiple entry feature discussed above, and which has been amply demonstrated to be absent from Agarwal. Claim 20 is therefore allowable over Agarwal.

Dependent claim 22.

The objection to claim 22 should be withdrawn since claim 22 depends on allowable claim 20.

Independent claim 28.

Agarwal does not anticipate claim 28 for at least the reason that Agarwal does not disclose a sequence of program instructions having a multiple-entry, single exit architecture as discussed above. Claim 28 is therefore allowable over Agarwal.

Dependent claim 30.

The objection to claim 30 should be withdrawn since claim 30 depends on allowable claim 28.

Independent claim 38.

Finally, Agarwal does not anticipate claim 38 for at least the reason that Agarwal does not disclose a trace having a multiple-entry, single exit architecture as discussed above. Claim 38 is therefore allowable over Agarwal..

Dependent claim 40.

The objection to claim 40 should be withdrawn since claim 40 depends on allowable claim 38.

Conclusion

In view of the above, it is clear that the Examiner erred in finally rejecting claims 1, 2, 16, 20, 28, 38 and 39 under 35 USC 102(e) as being anticipated by Agarwal. It is therefore respectfully requested that the Board reverse the Examiner and allow claims 1, 2, 16, 20, 28, 38 and 39.

The Examiner is invited to contact the undersigned at (202) 220-4323 to discuss any matter concerning this application. The Office is authorized to charge any fees related to this communication to Deposit Account No. 11-0600.

Respectfully submitted,

Dated: June 28, 2005

By: Will Curry
William E. Curry
Reg. No. 43,572

KENYON & KENYON
Attorneys for Intel Corporation
1500 K Street, N.W., Suite 700
Washington, D.C. 20005
Tel: (202) 220-4200
Fax:(202) 220-4201

APPENDIX

Claims on Appeal

1. Apparatus, comprising a memory entry storing a trace having a multiple-entry, single exit architecture.

2. The apparatus of claim 1, wherein the trace is a complex trace having multiple independent prefixes and a common, shared suffix.

16. A processing engine, comprising:

a front end stage storing blocks of instructions in a multiple-entry, single exit architecture when considered according to program flow, and

an execution unit in communication with the front end stage.

20. Apparatus, comprising a memory entry storing a sequence of program instructions as a trace, the instructions defining a program flow that progresses from any instruction therein to a last instruction in the memory entry and in which the trace has multiple separate prefixes.

28. A trace, comprising a sequence of program instruction assembled in order according to program flow, the sequence having a multiple-entry, single exit architecture.

29. The trace of claim 28, wherein the trace is a complex trace having multiple independent prefixes and a common, shared suffix.

38. Apparatus, comprising:

a memory having at least one memory entry; and

the at least one memory entry to store a trace having a multiple-entry, single exit architecture.

39. The apparatus of claim 38, wherein the trace is a complex trace having multiple independent prefixes and a common, shared suffix.

Application Ser. No. 09/608,624
Attorney Docket No. 2207/8609

RECEIVED
CENTRAL FAX CENTER
JUN 2 8 2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTORS: Stephan L. JOURDAN, et al.

SERIAL NO: 09/608,624

FILING DATE: June 30, 2000

TITLE: TRACE INDEXING VIA TRACE END ADDRESSES

ART UNIT: 2183

EXAMINER: Henry TSAI

Mail Stop Appeal Brief - Patents
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

SIR:

This is a brief in support of an appeal filed in the above-identified application.

I. Real Party in Interest

The real party in interest is the Assignee, Intel Corporation.

II. Related Appeals and Interferences

There are no other appeals or interferences known to Appellant, Appellant's legal representative, or Assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in this Appeal.

III. Status of Claims

The application as filed included claims 1-19. Claims 20-27 were added in an amendment filed September 4, 2003. Claims 8 and 21 were canceled and claims 28-37 were amended in an amendment filed April 8, 2004. Claims 31-37 were canceled and claims 38-43 were added in an amendment filed October 4, 2004. Claims 4-7, 9-15, 17-19, 23-27 and 41-43 are allowed. Claims 3, 22, 30 and 40 are objected to. Claims 1, 2, 16, 20, 28, 29, 38 and 39 have been finally rejected and are herein on appeal.

IV. Status of Amendments

The amendments filed March 4, 2005 in response to the final rejection mailed January 4, 2005 have been entered.

V. Summary of Claimed Subject Matter

(References are to the specification and drawings.)

Independent claim 1.

Apparatus, comprising a memory entry storing a trace having a multiple-entry, single exit architecture.

Explanation.

Embodiments of the present invention relate to an "extended block" architecture for instructions in a processor. The extended blocks are types of "traces." An extended block, according to embodiments of the present invention, may have multiple entry points but only a single exit point. (Page 2, lines 23-25.) Among other advantages, the traces provide for lessened redundancy, may be dynamically extended and may include shared blocks. (Page 3, lines 1-6.) The traces may be stored in a memory, such as a block cache 280. (Page 3, line 20 and FIG. 2.)

Examples of traces according to embodiments of the present invention are illustrated in FIGs. 4-6. As can be seen in these figures, the traces may have multiple

entry points but only a single exit point. For example, FIG. 4(c) shows a trace with two entry points IP_3 and IP_4, with a single exit point IP_1.

Independent claim 16.

A processing engine, comprising:

a front end stage storing blocks of instructions in a multiple-entry, single exit architecture when considered according to program flow, and

an execution unit in communication with the front end stage.

Explanation.

Instruction blocks having a multiple-entry, single exit architecture have been discussed above. As further discussed above, the instruction blocks may be stored in a memory, which in turn may be part of a processor front end stage. In particular, the memory may be a block cache 280 that is part of an extended block cache ("XBC") 220 that communicates with an execution unit. (Page 3, lines 7-8 , line 20, line 26 and FIG. 2.)

Independent claim 20.

Apparatus, comprising a memory entry storing a sequence of program instructions as a trace, the instructions defining a program flow that progresses from any instruction therein to a last instruction in the memory entry and in which the trace has multiple separate prefixes.

Explanation.

A trace may be a sequence of instructions in program order. (Page 1, lines 17-19.) According to embodiments of the present invention, the sequence may have different prefixes but a common suffix, so that all of the prefixes lead to a common last instruction. (Page 6, lines 5-7 and FIG. 4(c).

Independent claim 28.

A trace, comprising a sequence of program instructions assembled in order according to program flow, the sequence having a multiple-entry, single exit architecture.

Explanation.

A trace may be a sequence of instructions in program order. (Page 1, lines 17-19.) According to embodiments of the present invention, the sequence may have a multiple-entry, single exit architecture. (Page 2, lines 23-25.)

Independent claim 38.

Apparatus, comprising:

a memory having at least one memory entry; and

the at least one memory entry to store a trace having a multiple-entry, single exit architecture.

Explanation.

A trace may be a sequence of instructions in program order. (Page 1, lines 17-19.) According to embodiments of the present invention, the sequence may have a multiple-entry, single exit architecture. (Page 2, lines 23-25.) The traces may be stored in a memory, such as a block cache 280. (Page 3, line 20 and FIG. 2.)

VI. Grounds of Rejection to be Reviewed on Appeal

Claims 1, 2, 16, 20, 28, 38 and 39 were rejected under 35 USC 102(e) as being anticipated by Agarwal (US 5,966,541).

VII. Argument

Agarwal does not anticipate the rejected claims.

To anticipate a claim under § 102, a single prior art reference must identically disclose each and every claim element. See Lindeman Machinenfabrik v. American Hoist and Derrick, 730 F.2d 1452, 1458 (Fed. Cir. 1984). If any claimed element is absent from a prior art reference, it cannot anticipate the claim. See Rowe v. Dror, 112 F.3d 473, 478 (Fed. Cir. 1997). In view of the foregoing authority, the Applicant respectfully submits that the cited reference does not support the asserted rejection.

Independent claim 1.

Discussion begins with independent claim 1. Agarwal does not anticipate claim 1 for at least the reason that Agarwal does not disclose a memory ***entry*** that stores a trace having a multiple-entry, single exit architecture as recited in claim 1.

Agarwal contains no disclosure of a relationship between a memory entry and a trace. Instead, Agarwal's disclosure relates to software generally, and more particularly to an arrangement for "patching" faulty software, especially for the Year 2000 problem. In view of this, as an equivalent to the claimed memory entry, the Examiner can offer only "the space containing blocks 101, 102 and 103 as shown in Fig. 8" (final Office Action, item 5, lines 4-6). But this is hardly adequate; claim 1 requires that the trace be stored in a ***memory entry***, not merely memory generally. Agarwal does not disclose this subject matter. Agarwal describes no relationship between his blocks and memory, whether they fit in single entries or whether they are distributed across multiple memories. This isn't surprising since Agarwal is devoted to finding solutions to the Year 2000 problem – any relationship between his blocks 101-103 and memory entries is immaterial to the performance of his processes.

Further, it is observed that Agarwal does not mention traces at all. A trace is a specific code structure that occurs in processors. This structure cannot be found in Agarwal. Notwithstanding, the Examiner alleges that blocks 101-103 shown in FIG. 8 of Agarwal are equivalent to the claimed trace. See the final Office Action, lines 8-9: "... storing a trace (including blocks 101, 102 and 103 as shown in Fig. 8)". However, even assuming solely for purposes of argument that Agarwal's blocks 101, 102 and 103 correspond to the claimed trace, the blocks 101-103 at best have a *single-entry, single-exit* architecture. This is evident from FIG. 8 of Agarwal, an annotated version of which is shown below.

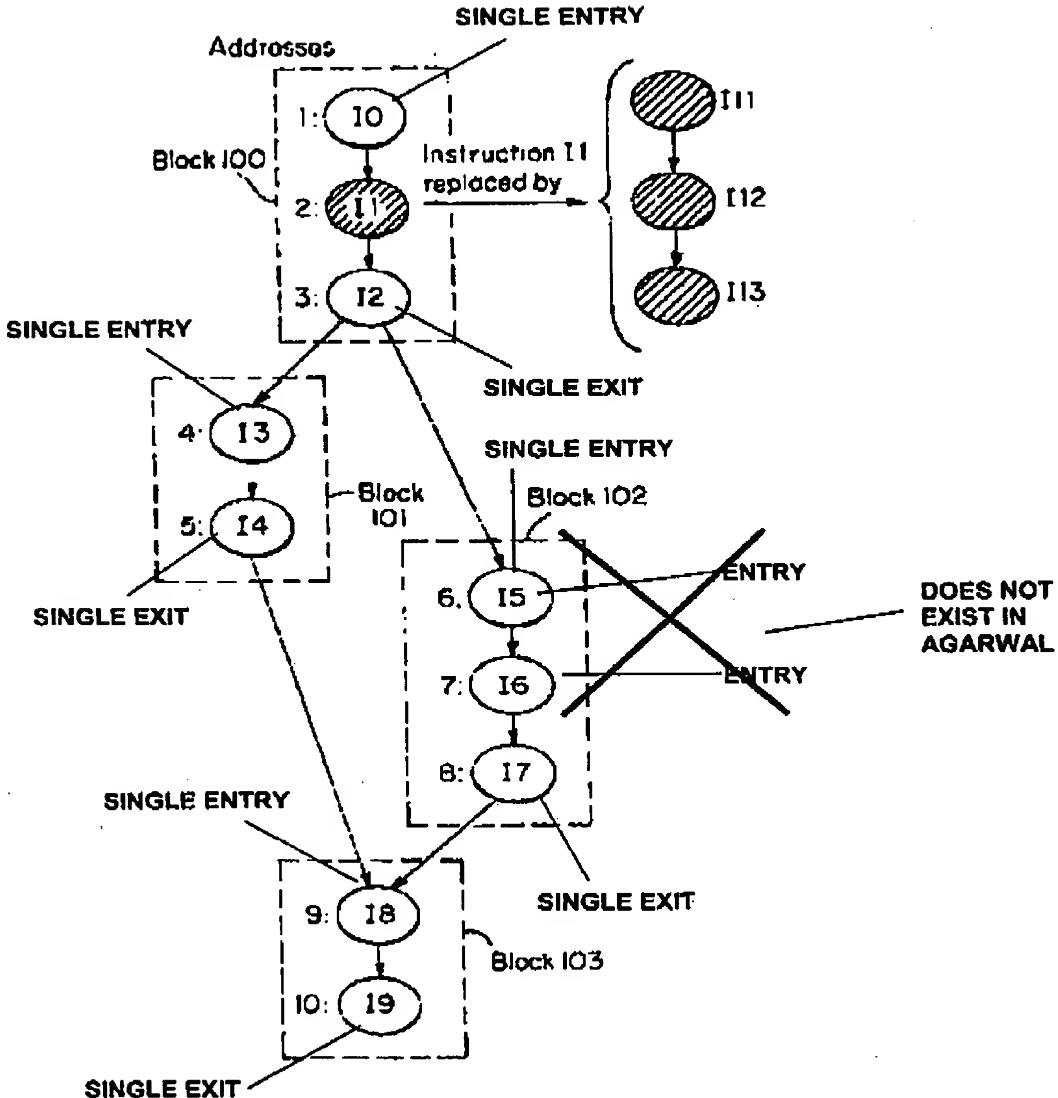


FIG. 8
(Agarwal, US 5,966,541)

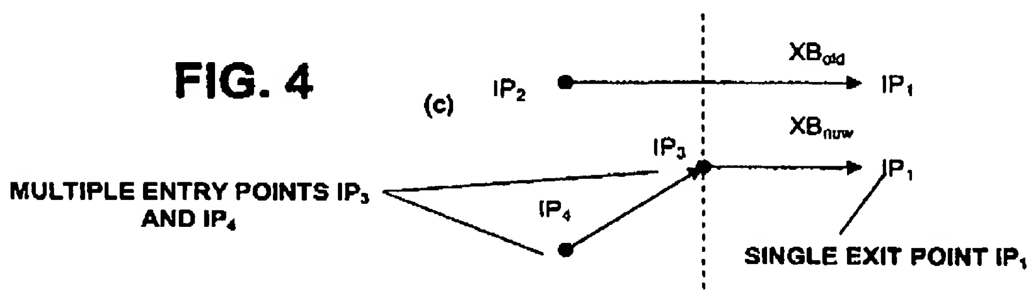
In the annotated FIG. 8, it can be seen that Agarwal dissects the code into separate blocks 100, 101, 102 and 103, each of which have a single entry, single exit architecture. Specifically, block 100 has a single entry point at I0 and a single exit point

at I2, block 101 has a single entry point at I3 and a single exit point at I4, and so on. Nowhere do the blocks exhibit a multiple entry structure, as illustrated hypothetically on block 102. Agarwal offers no teaching or suggestion to consider the code any other way.

In light of the above, it is apparent that the Examiner's strained effort to find multiple entry points in Agarwal's blocks 100-103 falls short. The Examiner refers specifically to "I2 to I3 and from I2 to I5" as multiple entry points (e.g., in the final Office Action, item 5, lines 9-10). However, under Agarwal's scheme, clearly I2 is not an entry, but an exit.

An annotated FIG. 4(c) of the present application, below, shows an example of the claimed multiple-entry, single-exit architecture by way of contrast to Agarwal's blocks 100-103. In FIG. 4(c), a trace has multiple entry points IP_3 and IP_4, and a single exit point IP_1.

FIG. 4



In view of the foregoing, claim 1 is allowable over Agarwal.

Dependent claims 2 and 3.

Rejected claim 2 depends on claim 1 and is therefore likewise allowable over Agarwal for at least the reasons discussed in connection with claim 1. Further, the objection to claim 3 should be withdrawn since claim 3 depends on allowable claim 1.

Independent claim 16.

Agarwal does not anticipate claim 16 for at least the reason that Agarwal does not disclose a front end stage storing blocks of instructions in a multiple-entry, single exit architecture when considered according to program flow. Moreover, Agarwal is completely silent as to a front end stage storing such blocks, and as to an execution unit in communication with the front end stage.

Agarwal's deficiencies concerning the multiple-entry, single exit architecture of embodiments of the present invention have been discussed above. As to the claimed front end stage in communication with an execution unit, Agarwal discloses only software structures with any specificity. Disclosure of hardware goes no deeper than "some memory" (col. 11, lines 36-38). Accordingly, Agarwal cannot meet the recitations of claim 16. Claim 16 is therefore allowable over Agarwal.

Independent claim 20.

Agarwal does not anticipate claim 20 for at least the reason that Agarwal does not disclose a trace with multiple separate prefixes as recited. This feature is analogous to the multiple entry feature discussed above, and which has been amply demonstrated to be absent from Agarwal. Claim 20 is therefore allowable over Agarwal.

Dependent claim 22.

The objection to claim 22 should be withdrawn since claim 22 depends on allowable claim 20.

Independent claim 28.

Agarwal does not anticipate claim 28 for at least the reason that Agarwal does not disclose a sequence of program instructions having a multiple-entry, single exit architecture as discussed above. Claim 28 is therefore allowable over Agarwal.

Dependent claim 30.

The objection to claim 30 should be withdrawn since claim 30 depends on allowable claim 28.

Independent claim 38.

Finally, Agarwal does not anticipate claim 38 for at least the reason that Agarwal does not disclose a trace having a multiple-entry, single exit architecture as discussed above. Claim 38 is therefore allowable over Agarwal..

Dependent claim 40.

The objection to claim 40 should be withdrawn since claim 40 depends on allowable claim 38.

Conclusion

In view of the above, it is clear that the Examiner erred in finally rejecting claims 1, 2, 16, 20, 28, 38 and 39 under 35 USC 102(e) as being anticipated by Agarwal. It is therefore respectfully requested that the Board reverse the Examiner and allow claims 1, 2, 16, 20, 28, 38 and 39.

The Examiner is invited to contact the undersigned at (202) 220-4323 to discuss any matter concerning this application. The Office is authorized to charge any fees related to this communication to Deposit Account No. 11-0600.

Respectfully submitted,

Dated: June 28, 2005

By: Will Curry
William E. Curry
Reg. No. 43,572

KENYON & KENYON
Attorneys for Intel Corporation
1500 K Street, N.W., Suite 700
Washington, D.C. 20005
Tel: (202) 220-4200
Fax:(202) 220-4201

APPENDIX

Claims on Appeal

1. Apparatus, comprising a memory entry storing a trace having a multiple-entry, single exit architecture.

2. The apparatus of claim 1, wherein the trace is a complex trace having multiple independent prefixes and a common, shared suffix.

16. A processing engine, comprising:

a front end stage storing blocks of instructions in a multiple-entry, single exit architecture when considered according to program flow, and

an execution unit in communication with the front end stage.

20. Apparatus, comprising a memory entry storing a sequence of program instructions as a trace, the instructions defining a program flow that progresses from any instruction therein to a last instruction in the memory entry and in which the trace has multiple separate prefixes.

28. A trace, comprising a sequence of program instruction assembled in order according to program flow, the sequence having a multiple-entry, single exit architecture.

29. The trace of claim 28, wherein the trace is a complex trace having multiple independent prefixes and a common, shared suffix.

38. Apparatus, comprising:

a memory having at least one memory entry; and

the at least one memory entry to store a trace having a multiple-entry, single exit architecture.

39. The apparatus of claim 38, wherein the trace is a complex trace having multiple independent prefixes and a common, shared suffix.

Application Ser. No. 09/608,624
Attorney Docket No. 2207/8609

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED
CENTRAL FAX CENTER
JUN 28 2005

INVENTORS: Stephan L. JOURDAN, et al.

SERIAL NO: 09/608,624

FILING DATE: June 30, 2000

TITLE: TRACE INDEXING VIA TRACE END ADDRESSES

ART UNIT: 2183

EXAMINER: Henry TSAI

Mail Stop Appeal Brief - Patents
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

SIR:

This is a brief in support of an appeal filed in the above-identified application.

I. Real Party in Interest

The real party in interest is the Assignee, Intel Corporation.

II. Related Appeals and Interferences

There are no other appeals or interferences known to Appellant, Appellant's legal representative, or Assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in this Appeal.

III. Status of Claims

The application as filed included claims 1-19. Claims 20-27 were added in an amendment filed September 4, 2003. Claims 8 and 21 were canceled and claims 28-37 were amended in an amendment filed April 8, 2004. Claims 31-37 were canceled and claims 38-43 were added in an amendment filed October 4, 2004. Claims 4-7, 9-15, 17-19, 23-27 and 41-43 are allowed. Claims 3, 22, 30 and 40 are objected to. Claims 1, 2, 16, 20, 28, 29, 38 and 39 have been finally rejected and are herein on appeal.

IV. Status of Amendments

The amendments filed March 4, 2005 in response to the final rejection mailed January 4, 2005 have been entered.

V. Summary of Claimed Subject Matter

(References are to the specification and drawings.)

Independent claim 1.

Apparatus, comprising a memory entry storing a trace having a multiple-entry, single exit architecture.

Explanation.

Embodiments of the present invention relate to an "extended block" architecture for instructions in a processor. The extended blocks are types of "traces." An extended block, according to embodiments of the present invention, may have multiple entry points but only a single exit point. (Page 2, lines 23-25.) Among other advantages, the traces provide for lessened redundancy, may be dynamically extended and may include shared blocks. (Page 3, lines 1-6.) The traces may be stored in a memory, such as a block cache 280. (Page 3, line 20 and FIG. 2.)

Examples of traces according to embodiments of the present invention are illustrated in FIGs. 4-6. As can be seen in these figures, the traces may have multiple

entry points but only a single exit point. For example, FIG. 4(c) shows a trace with two entry points IP_3 and IP_4, with a single exit point IP_1.

Independent claim 16.

A processing engine, comprising:

a front end stage storing blocks of instructions in a multiple-entry, single exit architecture when considered according to program flow, and

an execution unit in communication with the front end stage.

Explanation.

Instruction blocks having a multiple-entry, single exit architecture have been discussed above. As further discussed above, the instruction blocks may be stored in a memory, which in turn may be part of a processor front end stage. In particular, the memory may be a block cache 280 that is part of an extended block cache ("XBC") 220 that communicates with an execution unit. (Page 3, lines 7-8 , line 20, line 26 and FIG. 2.)

Independent claim 20.

Apparatus, comprising a memory entry storing a sequence of program instructions as a trace, the instructions defining a program flow that progresses from any instruction therein to a last instruction in the memory entry and in which the trace has multiple separate prefixes.

Explanation.

A trace may be a sequence of instructions in program order. (Page 1, lines 17-19.) According to embodiments of the present invention, the sequence may have different prefixes but a common suffix, so that all of the prefixes lead to a common last instruction. (Page 6, lines 5-7 and FIG. 4(c).

Independent claim 28.

A trace, comprising a sequence of program instructions assembled in order according to program flow, the sequence having a multiple-entry, single exit architecture.

Explanation.

A trace may be a sequence of instructions in program order. (Page 1, lines 17-19.) According to embodiments of the present invention, the sequence may have a multiple-entry, single exit architecture. (Page 2, lines 23-25.)

Independent claim 38.

Apparatus, comprising:

a memory having at least one memory entry; and

the at least one memory entry to store a trace having a multiple-entry, single exit architecture.

Explanation.

A trace may be a sequence of instructions in program order. (Page 1, lines 17-19.) According to embodiments of the present invention, the sequence may have a multiple-entry, single exit architecture. (Page 2, lines 23-25.) The traces may be stored in a memory, such as a block cache 280. (Page 3, line 20 and FIG. 2.)

VI. Grounds of Rejection to be Reviewed on Appeal

Claims 1, 2, 16, 20, 28, 38 and 39 were rejected under 35 USC 102(e) as being anticipated by Agarwal (US 5,966,541).

VII. Argument

Agarwal does not anticipate the rejected claims.

To anticipate a claim under § 102, a single prior art reference must identically disclose each and every claim element. See Lindeman Machinenfabrik v. American Hoist and Derrick, 730 F.2d 1452, 1458 (Fed. Cir. 1984). If any claimed element is absent from a prior art reference, it cannot anticipate the claim. See Rowe v. Dror, 112 F.3d 473, 478 (Fed. Cir. 1997). In view of the foregoing authority, the Applicant respectfully submits that the cited reference does not support the asserted rejection.

Independent claim 1.

Discussion begins with independent claim 1. Agarwal does not anticipate claim 1 for at least the reason that Agarwal does not disclose a memory ***entry*** that stores a trace having a multiple-entry, single exit architecture as recited in claim 1.

Agarwal contains no disclosure of a relationship between a memory entry and a trace. Instead, Agarwal's disclosure relates to software generally, and more particularly to an arrangement for "patching" faulty software, especially for the Year 2000 problem. In view of this, as an equivalent to the claimed memory entry, the Examiner can offer only "the space containing blocks 101, 102 and 103 as shown in Fig. 8" (final Office Action, item 5, lines 4-6). But this is hardly adequate; claim 1 requires that the trace be stored in a ***memory entry***, not merely memory generally. Agarwal does not disclose this subject matter. Agarwal describes no relationship between his blocks and memory, whether they fit in single entries or whether they are distributed across multiple memories. This isn't surprising since Agarwal is devoted to finding solutions to the Year 2000 problem – any relationship between his blocks 101-103 and memory entries is immaterial to the performance of his processes.

Further, it is observed that Agarwal does not mention traces at all. A trace is a specific code structure that occurs in processors. This structure cannot be found in Agarwal. Notwithstanding, the Examiner alleges that blocks 101-103 shown in FIG. 8 of Agarwal are equivalent to the claimed trace. See the final Office Action, lines 8-9: "... storing a trace (including blocks 101, 102 and 103 as shown in Fig. 8)". However, even assuming solely for purposes of argument that Agarwal's blocks 101, 102 and 103 correspond to the claimed trace, the blocks 101-103 at best have a *single-entry, single-exit* architecture. This is evident from FIG. 8 of Agarwal, an annotated version of which is shown below.

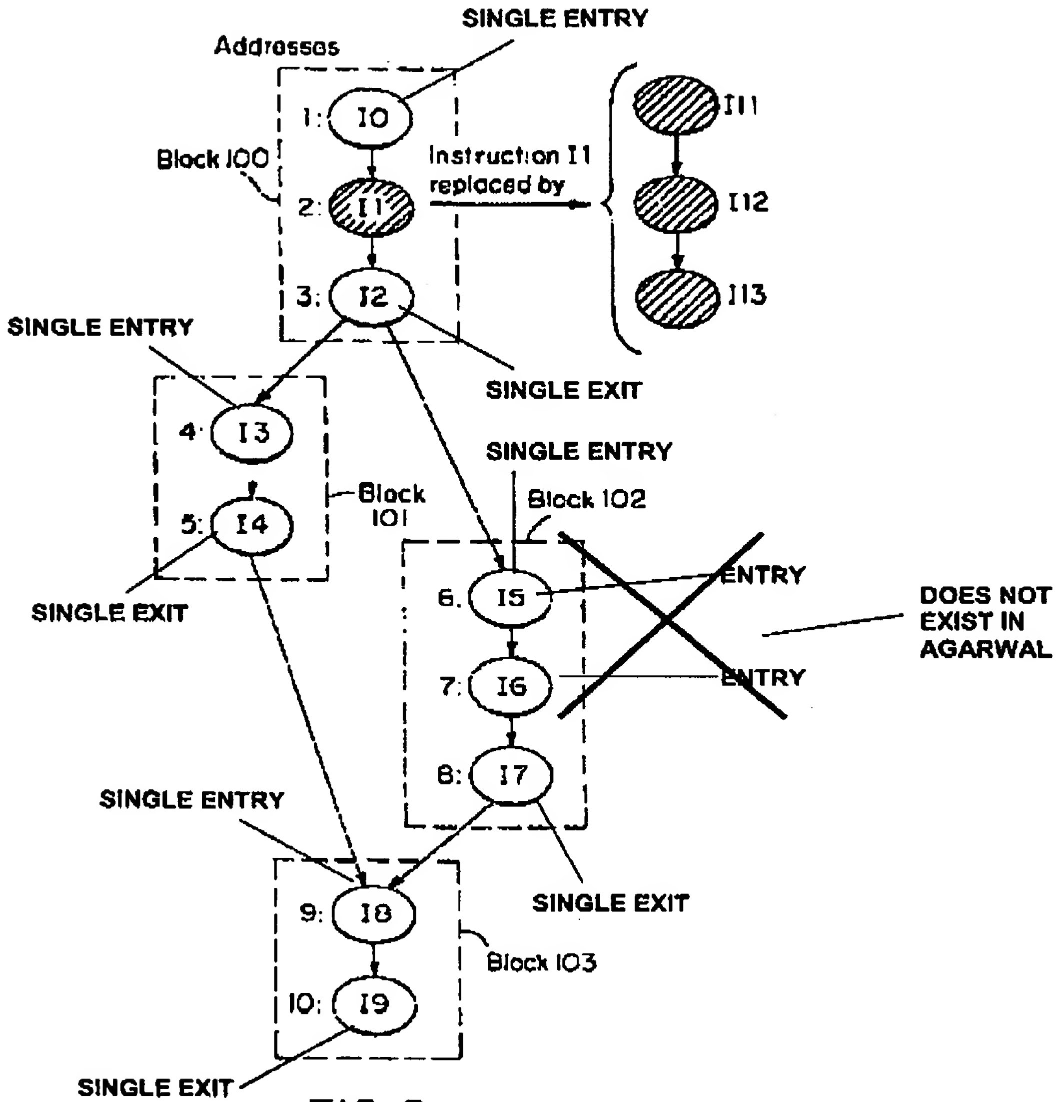


FIG. 8
(Agarwal, US 5,966,541)

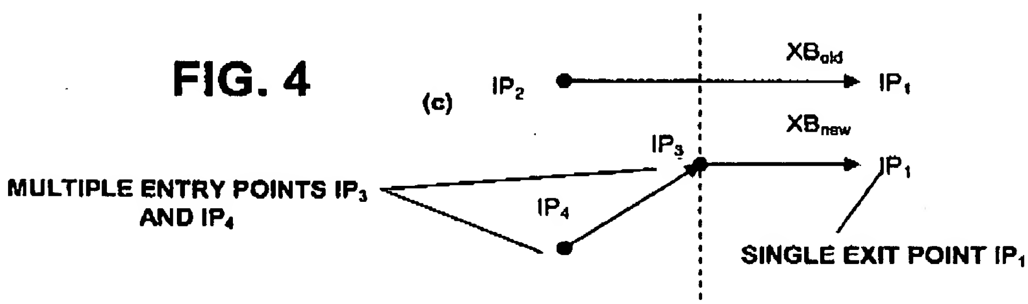
In the annotated FIG. 8, it can be seen that Agarwal dissects the code into separate blocks 100, 101, 102 and 103, each of which have a single entry, single exit architecture. Specifically, block 100 has a single entry point at I0 and a single exit point

at I2, block 101 has a single entry point at I3 and a single exit point at I4, and so on. Nowhere do the blocks exhibit a multiple entry structure, as illustrated hypothotically on block 102. Agarwal offers no teaching or suggestion to consider the code any other way.

In light of the above, it is apparent that the Examiner's strained effort to find multiple entry points in Agarwal's blocks 100-103 falls short. The Examiner refers specifically to "I2 to I3 and from I2 to I5" as multiple entry points (e.g., in the final Office Action, item 5, lines 9-10). However, under Agarwal's scheme, clearly I2 is not an entry, but an exit.

An annotated FIG. 4(c) of the present application, below, shows an example of the claimed multiple-entry, single-exit architecture by way of contrast to Agarwal's blocks 100-103. In FIG. 4(c), a trace has multiple entry points IP_3 and IP_4, and a single exit point IP_1.

FIG. 4



In view of the foregoing, claim 1 is allowable over Agarwal.

Dependent claims 2 and 3.

Rejected claim 2 depends on claim 1 and is therefore likewise allowable over Agarwal for at least the reasons discussed in connection with claim 1. Further, the objection to claim 3 should be withdrawn since claim 3 depends on allowable claim 1.

Independent claim 16.

Agarwal does not anticipate claim 16 for at least the reason that Agarwal does not disclose a front end stage storing blocks of instructions in a multiple-entry, single exit architecture when considered according to program flow. Moreover, Agarwal is completely silent as to a front end stage storing such blocks, and as to an execution unit in communication with the front end stage.

Agarwal's deficiencies concerning the multiple-entry, single exit architecture of embodiments of the present invention have been discussed above. As to the claimed front end stage in communication with an execution unit, Agarwal discloses only software structures with any specificity. Disclosure of hardware goes no deeper than "some memory" (col. 11, lines 36-38). Accordingly, Agarwal cannot meet the recitations of claim 16. Claim 16 is therefore allowable over Agarwal.

Independent claim 20.

Agarwal does not anticipate claim 20 for at least the reason that Agarwal does not disclose a trace with multiple separate prefixes as recited. This feature is analogous to the multiple entry feature discussed above, and which has been amply demonstrated to be absent from Agarwal. Claim 20 is therefore allowable over Agarwal.

Dependent claim 22.

The objection to claim 22 should be withdrawn since claim 22 depends on allowable claim 20.

Independent claim 28.

Agarwal does not anticipate claim 28 for at least the reason that Agarwal does not disclose a sequence of program instructions having a multiple-entry, single exit architecture as discussed above. Claim 28 is therefore allowable over Agarwal.

Dependent claim 30.

The objection to claim 30 should be withdrawn since claim 30 depends on allowable claim 28.

Independent claim 38.

Finally, Agarwal does not anticipate claim 38 for at least the reason that Agarwal does not disclose a trace having a multiple-entry, single exit architecture as discussed above. Claim 38 is therefore allowable over Agarwal..

Dependent claim 40.

The objection to claim 40 should be withdrawn since claim 40 depends on allowable claim 38.

Conclusion

In view of the above, it is clear that the Examiner erred in finally rejecting claims 1, 2, 16, 20, 28, 38 and 39 under 35 USC 102(e) as being anticipated by Agarwal. It is therefore respectfully requested that the Board reverse the Examiner and allow claims 1, 2, 16, 20, 28, 38 and 39.

The Examiner is invited to contact the undersigned at (202) 220-4323 to discuss any matter concerning this application. The Office is authorized to charge any fees related to this communication to Deposit Account No. 11-0600.

Respectfully submitted,

Dated: June 28, 2005

By: Will Curry
William E. Curry
Reg. No. 43,572

KENYON & KENYON
Attorneys for Intel Corporation
1500 K Street, N.W., Suite 700
Washington, D.C. 20005
Tel: (202) 220-4200
Fax:(202) 220-4201